UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/849,160	05/04/2001	Andre Kramer	CTX-032	9821

21323 7590 06/10/2004

TESTA, HURWITZ & THIBEAULT, LLP
HIGH STREET TOWER
125 HIGH STREET
BOSTON, MA 02110

EXAMINER	
CAO, DIEM K	

ART UNIT	PAPER NUMBER
2126	9

DATE MAILED: 06/10/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/849,160	KRAMER, ANDRE	
	Examiner	Art Unit	
	Diem K Cao	2126	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *04 May 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-33* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-33* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *12-3-2001*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-33 are presented for examination.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1-7, 11-18, 23 and 25-30 and 32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ims (U.S. 6,542,908 B1) in view of McDonald et al (U.S. 5,915,113).

4. **As to claim 1**, Ims teaches

- identifying one or more of the software components of the application program as a candidate for partitioning (a reference to a component ... executes; col. 10, lines 20-22, multiple beans ... when executing Netscape navigator; col. 11, line 56-61 and transforming components ... client/server computing environment; col. 7, lines 55-65);
- generating at the server system a plurality of new software components corresponding to one of the identified software component candidates (Split application ... accessed from any client; col. 7, line 66 – col. 8, line 6 and The archive builder ... a proxy; col. 10, lines 37-40);
- transmitting one of the plurality of new software components to the client system for execution at the client system (a proxy to be include din an archive file ...returned to the

client ... the bean proxy executes on the client; col. 10, lines 39-49), the new software component transmitted to the client system communicating with one of the other of the plurality of new software components at the server system when the application program is executed (When the bean's execution is invoked ... and bean instances such as bean 465; col. 10, line 60 – col. 11, line 4).

5. However, Ims does not explicitly teach generating a protocol to be used by the new software components to communicate with each other during an execution of the application program, and using the generated protocol to communicate between software components. Ims teaches the bean proxy and the generic client application are executed on the client system and the generic server application and bean component are executed on the server system. McDonald teaches the specific protocol is generated when partitioning a stand-alone application to be client/server application (A code generation ... to call the original part; col. 7, lines 20-25 and the client stub includes the appropriate middle-ware protocol; col. 7, lines 59-63).

6. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of Ims and McDonald because it provides a method for components on different computer system to communicate with each other during application execution.

7. **As to claim 2**, Ims teaches wrapping the identified software component candidate in one of the new software components generated at the server system (some code components reside

on the client machine and will execute at the client; col. 7, line 66 – col. 8, line 1 and a proxy to be included in an archive file ... some remote server; col. 10, lines 38-49).

8. **As to claim 3**, Ims teaches migrating the wrapped software component from one generated new software component to another generated new software component over the network (The parameter values passed ...operating system type and JVM level, respectively; col. 11, line 52 – col. 12, line 9).

9. **As to claim 4**, Ims does not explicitly teach the identified software component candidate is wrapped in the new software component that is transmitted to the client system for execution. However, Ims teaches come components are downloaded to the client system and execute at the client system (col. 7, line 66 – col. 8, line 1) and an archive file is generated to include all the information of how to access the component on the server which includes a bean proxy (col. 10, lines 38-49). It would have been obvious to one of ordinary skill in the art at the time the invention was made to motivate to send the identified software component to execute in the client because it reduces the burden for the server by execute some logic in the client system.

10. **As to claim 5**, Ims does not explicitly teach the new software components communicate with each other using a component protocol. Ims teaches the component on the client side (bean proxy) communicate with the bean instance on the server (bean) through the generic client application and the generic server application (col. 10, line 64 – col. 11). It would have been obvious the proxy bean and the bean instance utilize a component protocol for communication

because bean is one type of component and using component protocol would promote the reused of known technology in the new application.

11. **As to claim 6**, Ims does not explicitly teach the identified software component is wrapped in a new software component at the server system for execution at the server system. Ims teaches a bean is used to wrap legacy component code (col. 8, lines 61- 66), it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Ims into one application because it promotes the code reuse.

12. **As to claim 7**, Ims does not explicitly teach the new software components communicate with each other using an object protocol. Ims teaches the component on the client side (bean proxy) communicate with the bean instance on the server (bean) through the generic client application and the generic server application (col. 10, line 64 – col. 11). It would have been obvious a protocol is utilized for communication between client system and server system, and the object protocol could be used.

13. **As to claim 11**, Ims teaches the step of generating the plurality of software component candidate includes replicating external interfaces of the identified software component candidate for inclusion in each generated new software component (the remote proxy has an identical signature to the original bean; col. 11, lines 36-47).

14. **As to claim 12**, Ims teaches the steps of identifying one of the software components as a candidate for partitioning, generating the plurality of corresponding software components, and generating the protocol occur at runtime of the application program (the flow of messages ... a reference to a component is encountered ... a proxy for the actual bean instance; col. 10, lines 15-49).

15. **As to claim 13**, Ims teaches analyzing the identified software component candidate to determine whether to execute the identified software component candidate at the server system rather than partition the identified software component candidate for execution at the client system (a particular code component that is to be accessed can only be executed on a specific backend data server; col. 8, lines 45-53).

16. **As to claim 14**, Ims teaches

- analyzing a first software component to determine whether the first software component is to be partitioned (customize dynamically-generated components ... data server; col. 8, lines 35-53);
- if the first software component is to be partitioned

(a) dynamically generating a plurality of new software components corresponding to the first software component (Split application ... accessed from any client; col. 7, line 66 – col. 8, line 6 and The archive builder ... a proxy; col. 10, lines 37-40);

(b) transmitting one of the dynamically generated new software components to the client system for execution at the client system (a proxy to be include din an archive file ...returned

to the client ... the bean proxy executes on the client; col. 10, lines 39-49) and for communication with another of the generated new software components at the server system (When the bean's execution is invoked ... and bean instances such as bean 465; col. 10, line 60 – col. 11, line 4);

- otherwise, executing the first software component at the server and communicating with the client system (customize dynamically-generated components ... data server; col. 8, lines 35-53).

17. However, Ims does not teach a protocol to be used by the dynamically generated new software components for communicating with each other, and the dynamically generated protocol is used for communication between the new software components, and using a remote graphics protocol for communicating with the client when executing the first software component at the server. Ims teaches the bean proxy and the generic client application are executed on the client system and the generic server application and bean component are executed on the server system. McDonald teaches the specific protocol is generated when partitioning a stand-alone application to be client/server application (A code generation ... to call the original part; col. 7, lines 20-25 and the client stub includes the appropriate middle-ware protocol; col. 7, lines 59-63).

18. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of Ims and McDonald because it provides a method for

components on different computer system to communicate with each other during application execution.

19. **As to claim 15**, Ims teaches determining, if the first software component is to be partitioned, whether the first software component is to execute on the client system (some code components reside on the client machine and will execute at the client; col. 7, line 66 – col. 8).

20. **As to claim 16**, Ims does not explicitly teach the dynamically generated protocol is a component protocol when the first software component is to execute at the client system. Ims teaches the component on the client side (bean proxy) communicate with the bean instance on the server (bean) through the generic client application and the generic server application (col. 10, line 64 – col. 11). It would have been obvious the proxy bean and the bean instance utilize a component protocol for communication because bean is one type of component and using component protocol would promote the reused of known technology in the new application.

21. **As to claim 17**, Ims does not teach the dynamically generated protocol is an object protocol when the first software component is to execute at the server system. Ims teaches the component on the client side (bean proxy) communicate with the bean instance on the server (bean) through the generic client application and the generic server application (col. 10, line 64 – col. 11). It would have been obvious a protocol is utilized for communication between client system and server system, and the object protocol could be used.

22. **As to claim 18**, Ims teaches providing an integrated development environment in which to analyze the first software component to determine whether the first software component is to be partitioned and to partition the first software component if the first software component is to be partitioned (the proxy code has already been ... prior to runtime; col. 12, lines 26-29).

23. **As to claim 23**, it corresponds to the method claim of claim 1 except it a computer system claim.

24. **As to claim 25**, Ims teaches the first one of software components identified as a candidate is a Java Bean (a reference to a component such as a Java bean; col. 10, line 20-22).

As to claim 26, Ims teaches one of the new software components generated at the server system wraps the first one of software components identified as a candidate for partitioning (some code components reside on the client machine and will execute at the client; col. 7, line 66 – col. 8, line 1 and a proxy to be included in an archive file ... some remote server; col. 10, lines 38-49).

25. **As to claims 27-30**, see rejection of claims 4-7 above.

26. **As to claim 32**, Ims teaches each generated new software component corresponding to the identified software component candidate includes a copy of the external interfaces of the identified software component candidate (the remote proxy has an identical signature to the original bean; col. 11, lines 36-47).

27. Claims 8-10, 19-22, 24 and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ims (U.S. 6,542,908 B1) in view of McDonald et al (U.S. 5,915,113) further in view of Erlikh et al (U.S. 6,346,953 B1).

28. **As to claim 8**, Ims does not teach including in the identified software component candidate a description that facilitates identification of that software component as a candidate for partitioning. Erlikh teaches including in the identified software component candidate a description that facilitates identification of that software component as a candidate for partitioning (each screen-bound ... examined; col. 3, lines 15-24). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of Ims and Erlikh because it provides a method to convert a stand-alone application into distributed application by creating a user friendly graphical user interface from an existing un-friendly one.

29. **As to claim 9**, the step of identifying the software component as a candidate for partitioning includes determining that such software component has a software element that relates to a user interface (onscreen text, information field, font, color; col. 3, lines 5-8).

30. **As to claim 10**, Ims as modified teaches the software element is an external interface of the identified software component candidate (Introspection ... input property method, output property method and execution methods; col. 11, lines 43-47).

31. **As to claim 19**, Ims teaches

- generating a plurality of new software components corresponding to the software component of the application program (Split application ... accessed from any client; col. 7, line 66 – col. 8, line 6 and The archive builder ... a proxy; col. 10, lines 37-40);
- wrapping the software component with one of the new software components (some code components reside on the client machine and will execute at the client; col. 7, line 66 – col. 8, line 1 and a proxy to be included in an archive file ... some remote server; col. 10, lines 38-49);
- transmitting one of the new software components to the client system (a proxy to be include din an archive file ...returned to the client ... the bean proxy executes on the client; col. 10, lines 39-49);
- communicating with the new software component at the client system using a protocol when software component is executed (When the bean's execution is invoked ... and bean instances such as bean 465; col. 10, line 60 – col. 11, line 4).

32. However, Ims does not teach a user-interface software component, dynamically generated protocol is used when software component is executed, and communicating with the client system using a remote graphics protocol when the non-user-interface software component is executed. Ims teaches the bean proxy and the generic client application are executed on the client system and the generic server application and bean component are executed on the server system. McDonald teaches the specific protocol is generated when partitioning a stand-alone application to be client/server application (A code generation ... to call the original part; col. 7,

lines 20-25 and the client stub includes the appropriate middle-ware protocol; col. 7, lines 59-63). Erlikh teaches a user interface of an existing stand-alone application is analyzed and recreate into a graphical user interface so it can be used in the client/server system (abstract).

33. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of Ims, McDonald and Erlikh because it provides a method for components on different computer system to communicate with each other during application execution and a method to convert a stand-alone application into distributed application by creating a user friendly graphical user interface from an existing un-friendly one.

34. **As to claim 20**, see rejection of claim 16 above.

35. **As to claim 21**, see rejection of claim 17 above.

36. **As to claim 22**, Ims teaches building an application program comprised of the new software components using an integrated development environment (the proxy code has already been ... prior to runtime; col. 12, lines 26-29).

37. **As to claim 24**, see rejection of claim 9 above.

38. **As to claim 31**, see rejection of claim 8 above.

39. Claim 33 is rejected under 35 U.S.C. 103(a) as being unpatentable over McDonald et al (U.S. 5,915,113) in view of Ims (U.S. 6,542,908 B1).

40. **As to claim 33**, McDonald teaches a software component pallet listing a plurality of software components that are available for selection by an application program developer in constructing an application program (A visual building tool ... within one partition; col. 5, lines 62-67 and The user can begin moving objects ... on the visual display; col. 6, lines 22-32), and generating a protocol to be used by the new software components to communicate with each other (A code generation ... to call the original part; col. 7, lines 20-25 and the client stub includes the appropriate middle-ware protocol; col. 7, lines 59-63).

41. However, McDonald teaches a software component splitter generating a plurality of new software components from one of the software components listed by the software component pallet, and one of the new software components being generated for execution on a client system and another of the new software components being generated for execution on a server system. Ims teaches a software component splitter generating a plurality of new software components from one of the software components (Split application ... accessed from any client; col. 7, line 66 – col. 8, line 6 and The archive builder ... a proxy; col. 10, lines 37-40), and one of the new software components being generated for execution on a client system and another of the new software components being generated for execution on a server system (proxy, generic client application, generic server application; col. 10, line 60 – col. 11, line 4).

42. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of McDonald and Ims because it provides a method to create applications and components which can execute in a stand-alone mode or in a client/server mode without requiring a developer to change the applications or components.

Conclusion

43. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- Law et al teaches "Web-Enabling Legacy Application".
- Cole et al teaches "Using Java to Develop Web Based Tutorials".
- Sneed teaches "Encapsulating Legacy Software for Use in Client/Server Systems".

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Diem K Cao whose telephone number is (703) 305-5220. The examiner can normally be reached on Monday - Thursday, 9:00AM - 5:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (703) 305-9678. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Any response to this action should be mailed to:
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Diem Cao

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100